

L Number	Hits	Search Text	DB	Time stamp
1	197827	plugin or plug-in or (plug adj in) or module	USPAT	2004/02/17 10:14
2	145267	telephone or phone	USPAT	2004/02/17 10:15
3	7221	protocol with conver\$5	USPAT	2004/02/17 10:17
4	9877	(plugin or plug-in or (plug adj in) or module) same (telephone or phone)	USPAT	2004/02/17 10:17
5	108	(protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone))	USPAT	2004/02/17 10:18
6	32	((protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone))) and 379/\$.ccls.	USPAT	2004/02/17 10:18
7	95	((protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone))) and @ay<=1999	USPAT	2004/02/17 10:18
8	0	eigent and 379/\$.ccls.	USPAT	2004/02/17 11:51
9	60	eigen\$ and 379/\$.ccls.	USPAT	2004/02/17 11:52
10	15195	subscriber with (loop or line)	USPAT	2004/02/17 11:52
11	1742479	physical or structure	USPAT	2004/02/17 11:52
12	14	(eigen\$ and 379/\$.ccls.) and (subscriber with (loop or line)) and (physical or structure)	USPAT	2004/02/17 12:10
13	319	linear adj operator	USPAT	2004/02/17 12:12
14	229	(physical or structure) and (linear adj operator)	USPAT	2004/02/17 12:11
15	1	(subscriber with (loop or line)) and ((physical or structure) and (linear adj operator))	USPAT	2004/02/17 12:11
16	0	(eigen\$ and 379/\$.ccls.) and ((physical or structure) and (linear adj operator))	USPAT	2004/02/17 12:12
17	0	(eigen\$ and 379/\$.ccls.) and (linear adj operator)	USPAT	2004/02/17 12:12
18	1	(linear adj operator) and 379/\$.ccls.	USPAT	2004/02/17 12:12
19	12	(linear adj operator) and 324/\$.ccls.	USPAT	2004/02/17 12:21
20	6109	(linear or integral) with operator	USPAT	2004/02/17 12:22
21	22	(subscriber with (loop or line)) and ((linear or integral) with operator)	USPAT	2004/02/17 12:22
22	14	(physical or structure) and ((subscriber with (loop or line)) and ((linear or integral) with operator))	USPAT	2004/02/17 12:22

L Number	Hits	Search Text	DB	Time stamp
1	197827	plugin or plug-in or (plug adj in) or module	USPAT	2004/02/17 10:14
2	145267	telephone or phone	USPAT	2004/02/17 10:15
3	7221	protocol with conver\$5	USPAT	2004/02/17 10:17
4	9877	((plugin or plug-in or (plug adj in) or module) same (telephone or phone)	USPAT	2004/02/17 10:17
5	108	((protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone)))	USPAT	2004/02/17 10:18
6	32	((protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone))) and 379/\$.ccls.	USPAT	2004/02/17 10:18
7	95	((protocol with conver\$5) same ((plugin or plug-in or (plug adj in) or module) same (telephone or phone))) and @ay<=1999	USPAT	2004/02/17 10:18
8	0	eigent and 379/\$.ccls.	USPAT	2004/02/17 11:51
9	60	eigen\$ and 379/\$.ccls.	USPAT	2004/02/17 11:52
10	15195	subscriber with (loop or line)	USPAT	2004/02/17 11:52
11	1742479	physical or structure	USPAT	2004/02/17 11:52
12	14	((eigen\$ and 379/\$.ccls.) and (subscriber with (loop or line)) and (physical or structure))	USPAT	2004/02/17 12:10
13	319	linear adj operator	USPAT	2004/02/17 12:12
14	229	((physical or structure) and (linear adj operator)	USPAT	2004/02/17 12:11
15	1	((subscriber with (loop or line)) and ((physical or structure) and (linear adj operator)))	USPAT	2004/02/17 12:11
16	0	((eigen\$ and 379/\$.ccls.) and ((physical or structure) and (linear adj operator)))	USPAT	2004/02/17 12:12
17	0	((eigen\$ and 379/\$.ccls.) and (linear adj operator)	USPAT	2004/02/17 12:12
18	1	((linear adj operator) and 379/\$.ccls.	USPAT	2004/02/17 12:12
19	12	((linear adj operator) and 324/\$.ccls.	USPAT	2004/02/17 12:21
20	6109	((linear or integral) with operator	USPAT	2004/02/17 12:22
21	22	((subscriber with (loop or line)) and ((linear or integral) with operator)	USPAT	2004/02/17 12:22
22	14	((physical or structure) and ((subscriber with (loop or line)) and ((linear or integral) with operator)))	USPAT	2004/02/17 12:22
23	19	nguyen-duc.xp. and math\$	USPAT	2004/02/17 13:49
25	3	one-ended and 379/\$.ccls.	USPAT	2004/02/17 13:50
26	8	one-end\$2 and 379/\$.ccls.	USPAT	2004/02/17 13:51

09/572,527  
09/817,800

5,003,555

4

feedback filter by performing the linear signal transformation  $L_p$ .

This adaptive embodiment finally enables to further improve the already achieved transmission quality by adding a relatively simple non-adaptive post-detector to which the input signal of the symbol decision circuit is applied.

#### BRIEF DESCRIPTION OF THE DRAWING

The invention will be further explained hereinbelow with reference to the drawing, in which:

FIG. 1 shows a block diagram of a conceptual embodiment of a data transmission system in which the invention can be used;

FIG. 2 shows a functional discrete-time model of the system of FIG. 1 when conventional measures are employed;

FIG. 3 shows a functional discrete-time model of the system of FIG. 1 when the measures according to the invention are employed;

FIG. 4 shows a functional discrete-time model of an attractive embodiment of a system according to the invention; and

FIG. 5 shows a functional discrete-time model of an adaptive embodiment of a receiver of a system according to the invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In FIG. 1 a block diagram is shown of a system for data signal transmission with a data transmitter 1, a transmission channel 2 and a data receiver 3. The data transmitter 1 comprises a data signal source 10 for generating a data signal. This data signal is converted by an encoder 11 into a data signal which is transmitted through transmission channel 2 at a symbol rate  $1/T$ . The intersymbol interference (ISI) and noise developed during this transmission are combated in the data receiver 3. Thereto, data receiver 3 comprises an equalizer 30 of the decision feedback type which includes a feedforward filter 31 which is dimensioned for suppressing in the best way possible pre-cursive ISI and noise. On the basis of symbol decisions which are formed in a symbol decision circuit 32 a feedback filter 33 subsequently forms a cancelling signal for post-cursive ISI which is subtracted from the output signal of feedforward filter 31 by means of a difference circuit 34 for obtaining the input signal of symbol decision circuit 32. Finally, from the formed symbol decisions a decoder 35 forms a replica of the original data signal which is applied to a data signal sink 36.

To illustrate the problem for which the invention provides a solution, FIG. 2 shows a functional discrete-time model of the system of FIG. 1 when employing conventional measures. In the FIGS. 1 and 2 corresponding elements are denoted by the same reference symbols. The model of FIG. 2 is given for the case in which data signal source 10 generates a binary data signal and data transmitter 1 applies a ternary data signal to transmission channel 2.

A binary data signal  $d_k$  generated by data signal source 10 is converted by a non-linear part 12 of the encoder 11 into a likewise binary data signal  $a_k$  which, subsequently, by the linear part 13 of the encoder 11 is converted into a ternary data signal  $b_k$  to be applied to discrete-time transmission channel 2. To characterize the operation performed in this linear part 13 a partial-response polynomial  $g_r(D)$  can be used,  $D$  being a delay

operator representing the symbol interval  $T$ . Further details about these partial-response polynomials are to be found, for example, in the article "Partial-Response Signaling" by P. Kabal and S. Pasupathy, IEEE trans. Commun., Vol. COM-23, No. 9, pp. 921-934, Sept. 1975. For explaining the following description it should be observed that such polynomials generally have a relatively low order and also, apart from an otherwise unimportant scale factor, only have integral-valued coefficients. In the present case, for the purpose of illustration, the bipolar response 1-D for the polynomial  $g_r(D)$  is chosen such that

$$b_k = a_k - a_{k-1}. \quad (1)$$

The ternary data signal  $b_k$  is converted into an output signal  $r_k$  by the cascade arrangement of transmission channel 2 and feedforward filter 31 in FIG. 1 according to

$$r_k = (b * f^* w)_k + (n * w)_k, \quad (2)$$

where the symbol "\*" denotes the linear convolution-operator,  $f_k$  and  $w_k$  represent the discrete-time impulse responses of transmission channel 2 and feedforward filter 31, respectively, and  $n_k$  represents an additive discrete-time noise signal which is added by means of a summator 20.

With a proper dimensioning of the feedforward filter 31 of FIG. 1 it holds that the signal  $r_k$  contains virtually only post-cursive ISI. This implies that  $(f * w)_k$  can significantly differ from zero only for non-negative instants  $k$ . In the present system post-cursive ISI is combated by making feedback filter 33 have a causal impulse response  $p_k$  for which holds

$$p_k = \begin{cases} 0, & k \leq 0, \\ (f^* w)_k, & k \geq 1, \end{cases} \quad (3)$$

and applying to this feedback filter 33 the symbol decisions  $\hat{b}_k$  which are formed by decision circuit 32. As a result of the causal character of feedback filter 33 its output signal is at any instant  $k$  only determined by symbol decisions  $\hat{b}_{k-i}$  with  $i \geq 1$  that have already been formed. Under normal operating conditions these symbol decisions are correct, so that the output signal of the feedback filter 33 can be described as

$$(\hat{b} * p)_k = (b * p)_k. \quad (4)$$

The output signal  $\hat{b}_k$  of difference circuit 34 can now be described as

$$\hat{b}_k = r_k - (\hat{b} * p)_k. \quad (5)$$

In the case in which signal  $r_k$  only contains post-cursive ISI, this formula when utilizing formulas (2), (3) and (4) can be simplified to

$$\hat{b}_k = b_k + (n * w)_k = b_k + n'_k, \quad (6)$$

where  $n'_k$  represents the version of noise signal  $n_k$  that is attenuated in amplitude by feedforward filter 31. According to the latter formula, in the absence of error propagation, at the input of symbol decision circuit 32 an ISI-free estimate  $\hat{b}_k$  is formed of the data signal  $b_k$  at the output of data transmitter 1.

operands  
A ⊕ B = C  
↓ operator